Non-asymptotic analysis of an optimal algorithm for network-constrained averaging with noisy links

Nima Noorshams[1] and Martin J. Wainwright[1,2]

Departments of Statistics[2] and

Electrical Engineering & Computer Science[1],

University of California Berkeley,

{nshams, wainwrig}@eecs.berkeley.edu

Abstract

The problem of network-constrained averaging is to compute the average of a set of values distributed throughout a graph G using an algorithm that can pass messages only along graph edges. We study this problem in the noisy setting, in which the communication along each link is modeled by an additive white Gaussian noise channel. We propose a two-phase decentralized algorithm, and we use stochastic approximation methods in conjunction with the spectral graph theory to provide concrete (non-asymptotic) bounds on the mean-squared error. Having found such bounds, we analyze how the number of iterations $T_G(n;\delta)$ required to achieve mean-squared error δ scales as a function of the graph topology and the number of nodes n. Previous work provided guarantees with the number of iterations scaling inversely with the second smallest eigenvalue of the Laplacian. This paper gives an algorithm that reduces this graph dependence to the graph diameter, which is the best scaling possible.

I. Introduction

The problem of network-constrained averaging is to compute the average of a set of numbers distributed throughout a network, using an algorithm that is allowed to pass messages only along edges of the graph. Motivating applications include sensor networks, in which individual motes have limited memory and communication ability, and massive databases and server farms, in which memory constraints preclude storing all data at a central location. In typical applications, the average might represent a statistical estimate of some physical quantity (e.g., temperature, pressure etc.), or an intermediate quantity in a more complex algorithm (e.g., for distributed optimization). There is now an extensive literature on network-averaging, consensus problems, as well as distributed optimization and estimation (e.g., see the papers [7], [12], [10], [30], [20], [3], [4], [8], [23], [22]). The bulk of the earlier work has focused on the noiseless variant, in

which communication between nodes in the graph is assumed to be noiseless. A more recent line of work has studied versions of the problem with noisy communication links (e.g., see the papers [18], [15], [27], [2], [29], [19], [24] and references therein).

The focus of this paper is a noisy version of network-constrained averaging in which inter-node communication is modeled by an additive white Gaussian noise (AWGN) channel. Given this randomness, any algorithm is necessarily stochastic, and the corresponding sequence of random variables can be analyzed in various ways. The simplest question to ask is whether the algorithm is consistent—that is, does it compute an approximate average or achieve consensus in an asymptotic sense for a given fixed graph? A more refined analysis seeks to provide information about this convergence rate. In this paper, we do so by posing the following question: for a given algorithm, how does number of iterations required to compute the average to within δ-accuracy scale as a function of the graph topology and number of nodes n? For obvious reasons, we refer to this as the *network scaling* of an algorithm, and we are interested in finding an algorithm that has near-optimal scaling law.

The issue of network scaling has been studied by a number of authors in the noiseless setting, in which the communication between nodes is perfect. Of particular relevance here is the work of Benezit et al. [5], who in the case of perfect communication, provided a scheme that has essentially optimal message scaling law for random geometric graphs. A portion of the method proposed in this paper is inspired by their scheme, albeit with suitable extensions to multiple paths that are essential in the noisy setting. The issue of network scaling has also been studied in the noisy setting; in particular, past work by Rajagopal and Wainwright [27] analyzed a damped version of the usual consensus updates, and provided scalings of the iteration number as a function of the graph topology and size. However, our new algorithm has much better scaling than the method [27].

The main contributions of this paper are the development of a novel two-phase algorithm for network-constrained averaging with noise, and establishing the near-optimality of its network scaling. At a high level, the outer phase of our algorithm produces a sequence of iterates $\{\theta(\tau)\}_{\tau=0}^{\infty}$ based on a recursive linear update with decaying step size, as in stochastic approximation methods. The system matrix in this update is a time-varying and random quantity, whose structure is determined by the updates within the inner phase. These inner rounds are based on establishing multiple paths between pairs of nodes, and averaging along them simultaneously. By combining a careful analysis of the spectral properties of this random matrix with stochastic approximation theory, we prove that this two-phase algorithm computes a δ-accurate version of the average using a number of iterations that grows with the graph diameter (up to logarithmic factors).[1] As

[1]The graph diameter is the minimal number of edges needed to connect any two pairs of nodes in the graph.

we discuss in more detail following the statement of our main result, this result is optimal up to logarithmic factors, meaning that no algorithm can be substantially better in terms of network scaling.

The remainder of this paper is organized as follows. We begin in Section II with background and formulation of the problem. In Section III, we describe our algorithm, and state various theoretical guarantees on its performance. We then provide the proof of our main result in Section IV. Section V is devoted to some simulation results that confirm the sharpness of our theoretical predictions. We conclude the paper in Section VI.

Notation: For the reader's convenience, we collect here some notation used throughout the paper. The notation $f(n) = \mathcal{O}(g(n))$ means that there exists some constant $c \in (0, \infty)$ and $n_0 \in \mathbb{N}$ such $f(n) \leq cg(n)$ for all $n \geq n_0$, whereas $f(n) = \Omega(g(n))$ means that $f(n) \geq c'g(n)$ for all $n \geq n_0$. The notation $f(n) = \Theta(g(n))$ means that $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$. Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, we denote its ordered sequence of eigenvalues by $\lambda_1(A) \leq \lambda_2(A) \leq \ldots \leq \lambda_n(A)$ and also its l_2-operator norm by $|||A|||_2 = \sup_{\|v\|_2 = 1} \|Av\|_2$. Finally we use $\langle \cdot\,, \cdot \rangle$ to denote the Euclidean inner product.

II. Background and problem set-up

We begin in this section by introducing necessary background and setting up the problem more precisely.

A. Network-constrained averaging

Consider a collection $\{\theta_i(0),\ i = 1, \ldots, n\}$ of n numbers. In statistical settings, these numbers would be modeled as identically distributed (i.i.d.) draws from an unknown distribution $\mathbb{Q}$ with mean μ. In a centralized setting, a standard estimator for the mean is the sample average $\bar{\theta} := \frac{1}{n}\sum_{i=1}^{n} \theta_i(0)$. When all of the data can be aggregated at a central location, then computation of $\bar{\theta}$ is straightforward. In this paper, we consider the network-constrained version of this estimation problem, modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that consists of a vertex set $\mathcal{V} = \{1, \ldots, n\}$, and a collection of edges $\mathcal{E}$ joining pairs of vertices. For $i \in \mathcal{V}$, we view each measurement $\theta_i(0)$ as associated with vertex i. (For instance, in the context of sensor networks, each vertex would contain a mote and collect observations of the environment.) The edge structure of the graph enforces communication constraints on the processing: in particular, the presence of edge (i, j) indicates that it is possible for sensors i and j to exchange information via a noisy communication channel. Conversely, sensor pairs that are *not* joined by an edge are not permitted to communicate directly.[2] Every node has a synchronized internal clock, and acts at discrete times $t = 1, 2, \cdots$. For any given pair of

[2]Moreover, since the edges are undirected, there is no difference between edge (i, j) and (j, i); moreover, we exclude self-edges, meaning that $(i, i) \notin \mathcal{E}$ for all $i \in \mathcal{V}$.

sensors $(i,j) \in \mathcal{E}$, we assume that the message sent from i to j is perturbed by an independent identically distributed $N(0,\sigma^2)$ variate. Although this additive white Gaussian noise (AWGN) model is more realistic than a noiseless model, it is conceivable (as pointed out by one of the reviewers) that other stochastic channel models might be more suitable for certain types of sensor networks, and we leave this exploration for future research.

Given this set-up, of interest to us are stochastic algorithms that generate sequences $\{\theta(t)\}_{t=0}^{\infty}$ of iterates contained within $\mathbb{R}^n$, and we require that the algorithm be *graph-respecting*, meaning that in each iteration, it is allowed to send at most one message for each direction of every edge $(i,j) \in \mathcal{E}$. At time t, we measure the distance between $\theta(t)$ and the desired average $\bar{\theta}$ via the average (per node) mean-squared error, given by

$$\mathrm{MSE}(\theta(t)) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[(\theta_i(t) - \bar{\theta})^2]. \tag{1}$$

In this paper, our goal is for every node to compute the average $\bar{\theta}$ up to an error tolerance δ. In addition, we require almost sure consensus among nodes, meaning

$$\mathbb{P}[\theta_i(t) = \theta_j(t) \quad \forall\, i,j = 1,2,\cdots,n] \to 1 \quad \text{as } t \to \infty.$$

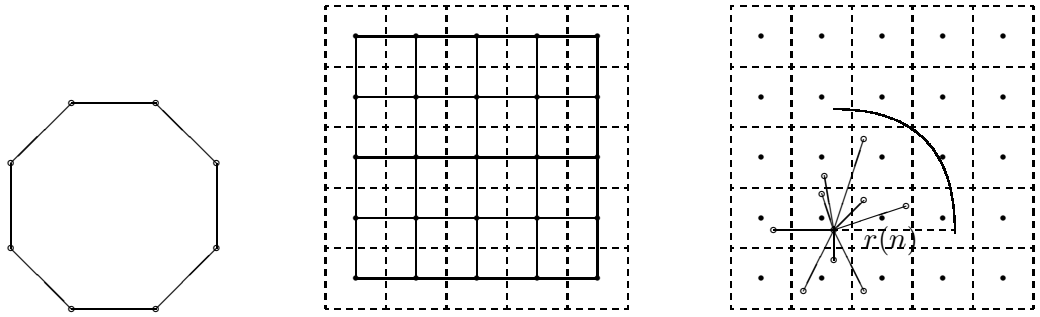
Our primary goal is in characterizing the rate of convergence as a function of the graph topology and the number of nodes, to which we refer as the *network-scaling function* of the algorithm. More precisely, in order to study this network scaling, we consider sequences of graphs $\{\mathcal{G}_n\}$ indexed by the number of nodes n. For any given algorithm (defined for each graph $\mathcal{G}_n$) and a fixed tolerance parameter $\delta > 0$, our goal is to determine bounds on the quantity

$$T_{\mathcal{G}}(n;\delta) := \inf\big\{t = 1,2,\ldots \mid \mathrm{MSE}(\theta(t)) \le \delta\big\}. \tag{2}$$

Note that $T_{\mathcal{G}}(n;\delta)$ is a stopping time, given by the smallest number of iterations required to obtain mean-squared error less than δ on a graph of type $\mathcal{G}$ with n nodes.

B. Graph topologies

Of course, the question that we have posed will depend on the graph type, and this paper analyzes three types of graphs, as shown in Figure 1. The first two graphs have regular topologies: the single cycle graph in panel (a) is degree two-regular, and the two-dimensional grid graph in panel (b) is degree four-regular. In addition, we also analyze an important class of random graphs with irregular topology, namely the class of random geometric graphs. As illustrated in Figure 1(c), a random geometric graph (RGG) in the plane



(a) Single cycle. (b) Two-dimensional grid. (c) Random geometric graph.

Fig. 1. Illustration of graph topologies. (a) A single cycle graph. (b) Two-dimensional grid with four-nearest-neighbor connectivity. (c) Illustration of a random geometric graph (RGG). Two nodes are connected if their distance is less than $r(n)$. The solid circles represent the center of squares.

is formed according by placing n nodes uniformly at random in the unit square $[0,1] \times [0,1]$, and the connecting two nodes if their Euclidean distance is less than some radius $r(n)$. It is known that an RGG will be connected with high probability as long as $r(n) = \Omega(\sqrt{\frac{\log n}{n}})$; see Penrose [26] for discussion of this and other properties of random geometric graphs.

A key graph-theoretic parameter relevant to our analysis is the *graph diameter*, denoted by $D_n = \mathrm{diam}(\mathcal{G}_n)$. The path distance between any pair of nodes is the length of the shortest path joining them in the graph, and by definition, the graph diameter is the maximum path distance taken over all node pairs in the graph. It is straightforward to see that $D_n = \Theta(n)$ for the single cycle graph, and that $D_n = \Theta(\sqrt{n})$ for the two-dimensional grid. For a random geometric graph with radius chosen to ensure connectivity, it is known that $D_n = \Theta\left(\sqrt{\frac{n}{\log n}}\right)$.

Finally, in order to simplify the routing problem explained later, we divide the unit square into subregions (squares) of side length $\sqrt{\frac{1}{n}}$ in case of grid, and for some constant $c > 0$, of side length $\sqrt{c\frac{\log n}{n}}$ in case of RGG. We assume that each node knows its location and is aware of the center of these m^2 subregions namely (x_i, y_j) $i, j = 1, 2, \cdots, m$, where $m = \sqrt{n}$ for the regular grid, and $m = \sqrt{\frac{n}{c \log n}}$ for the RGG. As a convention, we assume that (x_1, y_1) is the left bottom square, to which we refer to as the first square. By construction, in a regular grid, each square will contain one and only one node which is located at the center of the square. From known properties of RGGs [26], [17], each of the given subregions will contain at least one node with high probability (w.h.p.). Moreover, an RGG is regular w.h.p, meaning that each square contains $\Theta(\log n)$ nodes (see Lemma 1 in the paper [12]). Accordingly, in the remainder of the paper, we assume without loss of generality that any given RGG is regular. Note that by construction, the transmission radius $r(n)$ is selected so that each node in each square is connected to every other node in four adjacent squares.

III. Algorithm and its properties

In this section we state our main result which is followed by a detailed description of the proposed algorithm.

A. Theoretical guarantees

Our main result guarantees the existence of a graph-respecting algorithm with desirable properties. Recall the definition of the graph respecting scheme, as well as the definition of our AWGN channel model given in Section II. In the following statement, the quantity c_0 denotes a universal constant, independent of n, δ, and σ^2.

Theorem 1. *For the communication model in which each link is an AWGN channel with variance σ^2, there is a graph-respecting algorithm such that:*

a) Nodes almost surely reach a consensus. More precisely, we have

$$\theta(t) \xrightarrow{a.s.} \widetilde{\theta}\,\vec{1} \quad \text{as } t \to \infty, \tag{3}$$

for some $\widetilde{\theta} \in \mathbb{R}$.

b) After $T = T_{\mathcal{G}}(n;\delta)$ iterations, the algorithm satisfy the following bounds on the $\mathrm{MSE}(\theta(T))$:

i) For fixed tolerance $\delta > 0$ sufficiently small, we have $\mathrm{MSE}(\theta(T)) \leq 3\,\sigma^2\delta$ after

$$T_{\mathrm{cyc}}(n;\delta) \;\leq\; c_0\, n\, \max\left\{\frac{1}{\delta}\log\frac{1}{\delta}\,,\; \frac{\mathrm{MSE}(\theta(0))}{\sigma^2\delta^2}\right\}$$

iterations for a single cycle graph.

ii) For fixed tolerance $\delta > 0$ sufficiently small, we have $\mathrm{MSE}(\theta(T)) = \mathcal{O}\left(\sigma^2\delta\right)$ after

$$T_{\mathrm{grid}}(n;\delta) \;\leq\; c_0\, \sqrt{n}\, \max\left\{\frac{1}{\delta}\log\frac{1}{\delta}\,,\; \frac{\mathrm{MSE}(\theta(0))}{\sigma^2\delta^2}\right\}$$

iterations for the regular grid in two dimensions.

iii) Assume that $\delta = \frac{\widetilde{\delta}}{(\log n)^2}$, for some fixed $\widetilde{\delta}$ sufficiently small. Then we have $\mathrm{MSE}(\theta(T)) = \mathcal{O}\left(\sigma^2\widetilde{\delta}\right)$ after

$$T_{\mathrm{RGG}}(n;\delta) \;\leq\; c_0\, \sqrt{n(\log n)^3}\, \max\left\{\frac{1}{\widetilde{\delta}}\log\frac{(\log n)^2}{\widetilde{\delta}}\,,\; \frac{\mathrm{MSE}(\theta(0))}{\sigma^2\widetilde{\delta}^2}\right\}$$

iterations for a regular random geometric graph.

Here c_0 is some constant independent of n, δ, and σ^2, whose value may change from line to line.

Remarks: A few comments are in order regarding the interpretation of this result. First, it is worth mentioning that the quality of the different links does not have to be the same. Similar arguments apply to the case where noises have different variances. Second, although nodes almost surely reach a consensus, as guaranteed in part (a), this consensus value is not necessarily the same as the sample mean $\bar{\theta}$. The choice of $\widetilde{\theta}$ is intentional to emphasize this point. However, as guaranteed by part (b), this consensus value is within $\sigma^2\delta$ distance of the actual sample mean. Since the sample mean itself represents a noisy estimate of some underlying population quantity, there is little point to computing it to arbitrary accuracy. Third, it is worthwhile comparing part (b) with previous results on network scaling in the noisy setting. Rajagopal and Wainwright [27] analyzed a simple set of damped updates, and showed that $T_{\mathrm{cyc}}(n;\delta) = \mathcal{O}\left(n^2\right)$ for the single cycle, and that $T_{\mathrm{grid}}(n) = \mathcal{O}\left(n\right)$ for the two-dimensional grid. By comparison, the algorithm proposed here and our analysis thereof has removed factors of n and $\sqrt{n}$ from this scaling.

B. Optimality of the results

As we now discuss, the scalings in Theorem 1 are optimal for the cases of cycle and grid and near-optimal (up to logarithmic factor) for the case of RGG. In an adversarial setting, any algorithm needs at least $\Omega(D_n)$ iterations, where D_n denotes the graph diameter, in order to approximate the average; otherwise, some node will fail to have any information from some subset of other nodes (and their values can be set in a worst-case manner). Theorem 1 provides upper bounds on the number of iterations that, at most, are within logarithmic factors of the diameter, and hence are also within logarithmic factors of the optimal latency scaling law. For the graphs given here, the scalings are also optimal in a non-adversarial setting, in which $\{\theta_i(0)\}_{i=1}^n$ are modeled as chosen i.i.d. from some distribution. Indeed, for a given node $j \in \mathcal{V}$, and positive integer t, we let $\mathcal{N}(j;t)$ denote the depth t neighborhood of j, meaning the set of nodes that are connected to j by a path of length at most t. We then define the graph spreading function $\psi_{\mathcal{G}}(t) = \min_{j\in\mathcal{V}} |\mathcal{N}(j;t)|$. Note that the function $\psi_{\mathcal{G}}$ is non-decreasing, so that we may define its inverse function $\psi_{\mathcal{G}}^{-1}(s) = \inf\{t \mid \psi_{\mathcal{G}}(t) \leq s\}$. As some examples:

- for a cycle on n nodes, we have $\psi_{\mathcal{G}}(t) = 2t$, and hence $\psi_{\mathcal{G}}^{-1}(s) = s/2$.
- for a n-grid in two dimensions, we have the upper bound $\psi_{\mathcal{G}}(t) \leq 2t^2$, and hence the lower bound $\psi_{\mathcal{G}}^{-1}(s) \geq \sqrt{\frac{s}{2}}$.
- for a random geometric graph (RGG), we have the upper bound $\psi_{\mathcal{G}}(t) = \Theta(t^2 \log n)$, which implies the lower bound $\psi_{\mathcal{G}}^{-1}(s) = \Theta\left(\sqrt{\frac{s}{\log n}}\right)$

After t steps, a given node can gather the information of at most $\psi_{\mathcal{G}}(t)$ nodes. For the average based on $\psi_{\mathcal{G}}(t)$ nodes to be comparable to $\bar{\theta}$, we require that $\psi_{\mathcal{G}}(t) = \Omega(n)$, and hence the iteration number t should be at least $\Omega(\psi_{\mathcal{G}}^{-1}(n))$. For the three graphs considered here, this leads to the same conclusion, namely that

$\Omega(D_n)$ iterations are required. We note also that using information-theoretic techniques, Ayaso et al. [1] proved a lower bound on the number of iterations for a general graph in terms of the Cheeger constant [9]. For the graphs considered here, the Cheeger constant is of the order of the diameter.

C. Description of algorithm

We now describe the algorithm that achieves the bounds stated in Theorem 1. At the highest level, the algorithm can be divided into two types of phases: an inner phase, and an outer phase. The outer phase produces a sequence of iterates $\{\theta(\tau)\}$, where $\tau = 0, 1, 2, \ldots$ is the outer time scale parameter. By design of the algorithm, each update of the outer parameters requires a total of M message-passing rounds (these rounds corresponding to the inner phase), where in each round the algorithm can pass at most two messages per edge (one for each direction). To put everything in a nutshell, the algorithm is based on establishing multiple routes, averaging along them in an inner phase and updating the estimates based on the noisy version of averages along routes in an outer phase. Consequently, if we use the estimate $\theta(\tau)$, then in the language of Theorem 1, it corresponds to $T = M\tau$ rounds of message-passing. Our goal is to establish upper bounds on T that guarantee the MSE is $\mathcal{O}(\sigma^2\delta)$. Figure 2 illustrates the basic operations of the algorithm.

Two-phase algorithm for distributed consensus:

- Inner phase:
  - Deciding the averaging direction
  - Choosing the head nodes
  - Establishing the routes
  - Averaging along the routes
- Outer phase:
  - Based on the averages along the routes, update the estimates according to

$$\theta(\tau+1) = \theta(\tau) - \epsilon(\tau)\big\{L(\tau)\theta(\tau) + v(\tau)\big\}$$

Fig. 2: Basic operations of a two-phase algorithm for distributed consensus.

1) Outer phase: In the outer phase, we produce a sequence of iterates $\{\theta(\tau)\}_{\tau=1}^{\infty}$ according to the recursive update

$$\theta(\tau+1) = \theta(\tau) - \epsilon(\tau)\big\{L(\tau)\theta(\tau) + v(\tau)\big\}. \tag{4}$$

Here $\{\epsilon(\tau)\}_{\tau=1}^{\infty}$ is a sequence of positive decreasing stepsizes. For a given precision, δ, we set $\epsilon(\tau) = 1/(\frac{1}{\delta} + \tau)$. For each τ, the quantity $L(\tau) \in \mathbb{R}^{n\times n}$ is a random matrix, whose structure is determined by the

inner phase, and $v(\tau) \in \mathbb{R}^n$ is an additive Gaussian term, whose structure is also determined in the inner phase. As will become clear in the sequel, even though L and v are dependent, they are both independent of θ. Moreover, given L, the random vector v is Gaussian with bounded variance.

2) Inner phase: The inner phase is the core of the algorithm and it involves a number of steps, as we describe here. We use $s = 1, 2, \ldots, M$ to index the iterations within any inner phase, and use $\{\gamma(s)\}_{s=1}^{M}$ to denote the sequence of inner iterates within $\mathbb{R}^n$. For the inner phase corresponding to outer update from $\theta(\tau) \to \theta(\tau+1)$, the inner phase takes the initialization $\gamma(1) \leftarrow \theta(\tau)$, and then reduces as output $\gamma(M) \to \theta(\tau+1)$ to the outer iteration. In more detail, the inner phase can be broken down into three steps, which we now describe in detail.

a) Step 1, deciding the averaging direction: The first step is to choose a direction in which to perform averaging. In a single cycle graph, since left and right are viewed as the same, there is only one choice, and hence nothing to be decided. In contrast, the grid or RGG graphs require a decision-making phase, which proceeds as follows. One node in the first (bottom left) square, wakes up and chooses uniformly at random to send in the horizontal or vertical direction. We code this decision using the random variable $\zeta \in \{-1, 1\}$, where $\zeta = -1$ (respectively $\zeta = +1$) represents the horizontal (respectively vertical) direction. To simplify matters, we assume in the remainder of this description that the averaging direction is horizontal, with the modifications required for vertical averaging being standard.

b) Step 2, choosing the head nodes: This step applies only to the grid and RGG graphs. Given our assumption that the node in the first square has chosen the horizontal direction, it then passes a token message to a randomly selected node in the above adjacent square. The purpose of this token is to determine which node (referred to as the head node) should be involved in establishing the route passing through the given square. After receiving the token, the receiving node passes it to another randomly selected node in the above adjacent square and so on. Note that in the special case of grid, there is only one node in each square, and so no choices are required within squares. After m rounds, one node in each square $(x_1, y_j), j = 1, 2, \cdots, m$ ($(x_i, y_1), i = 1, 2, \cdots, m$) receives the token, as illustrated in Figure 3. Note that again in a single cycle graph, there is nothing to be decided, since the direction and head nodes are all determined.

c) Step 3, establishing routes and averaging: In this phase, each of head nodes establishes a horizontal path, and then perform averaging along the path, as illustrated in Figure 3(b). This part of algorithm involves three substeps, which we now describe in detail.

- For $j = 1, 2, \cdots, m$, each head node s_{1j} selects a node s_{2j} uniformly at random (u.a.r.) from within the right adjacent square, and passes to it the quantity $\gamma_{1j}(1)$. Given the Gaussian noise model, node s_{2j} then

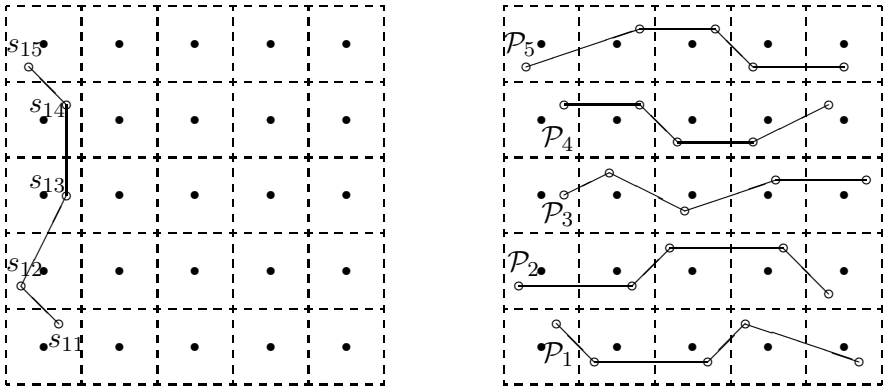


Fig. 3. (a) The node labeled s_{11} in the first square, chooses the horizontal direction for averaging ($\zeta = -1$); it passes the token vertically to inform other nodes to average horizontally. Nodes who receive the token pass it to another node in the above adjacent square. (b) The head nodes s_{1j} $j = 1, 2, \cdots$, as determined in the first step, establish routes horizontally ($\mathcal{P}_j$, $j = 1, 2, \cdots, m$) and then average along these paths.

receives the quantity

$$\widetilde{\gamma}_{1j}(1) = \gamma_{1j}(1) + v_{1j}, \quad \text{where } v_{1j} \sim N(0, \sigma^2),$$

and then updates its own local variable as $\gamma_{2j}(2) = \gamma_{2j}(1) + \widetilde{\gamma}_{1j}(1)$. We then iterate this same procedure—that is, node s_{2j} selects another s_{3j} u.a.r. from its right adjacent square, and passes the message $\gamma_{2j}(2)$. Overall, at round i of this update procedure, we have

$$\gamma_{(i+1)j}(i+1) = \gamma_{(i+1)j}(i) + \widetilde{\gamma}_{ij}(i),$$

where $\widetilde{\gamma}_{ij}(i) = \gamma_{ij}(i) + v_{ij}$, and $v_{ij} \sim N(0, \sigma^2)$. At the end of round m, node s_{mj} can compute a noisy version of the average along the path $\mathcal{P}_j : s_{1j} \to s_{2j} \to \cdots \to s_{mj}$, in particular via the rescaled quantity

$$\eta_j := \frac{\gamma_{mj}(m)}{m} = \frac{1}{m} \sum_{i=1}^{m} \theta_{s_{ij}}(t) + v_j \quad j = 1, 2, \cdots, m.$$

Here the variable $v_j \sim \mathcal{N}(0, \frac{\sigma^2}{m})$, since the noise variables associated with different edges are independent.

- At this point, for each $j = 1, 2, \ldots, m$, each node s_{mj} which has the noisy version, η_j, of the path average along route $\mathcal{P}_j$; can share this information with other nodes in the path by sending η_j back to the head node. A naive way to do this is as follows: node s_{mj} makes m copies of η_j—namely, $\eta_j^{(l)} = \eta_j$, $l = 1, 2, \cdots, m$—and starts transmitting one copy at a time back to the head node. Nodes along the path simply forward what they receive, so that after $m - i + m - 1$ time steps, node s_{ij} receives m noisy copies of the average, $\widetilde{\eta}_{ij}^{(l)} = \eta_j^{(l)} + v_{ij}^{(l)}$ where $v_{ij}^{(l)} \sim \mathcal{N}(0, (m-i)\sigma^2)$. Averaging the m copies, node s_{ij}

can compute the quantity

$$\gamma_{ij}(3m-i-1) := \frac{1}{m}\sum_{l=1}^{m}\widetilde{\eta}_{ij}^{(l)} \;=\; \frac{1}{m}\sum_{l=1}^{m}\theta_{s_{lj}}(\tau) + w_{ij},$$

where $w_{ij} = v_j + \frac{1}{m}\sum_{l=1}^{m} v_{ij}^{(l)}$. Since the noise on different links and different time steps are independent Gaussian random variables, we have $w_{ij} \sim \mathcal{N}(0, \sigma_i^2)$, with

$$\sigma_i^2 = \frac{1}{m}\sigma^2 + (1-\frac{i}{m})\sigma^2 = (1-\frac{(i-1)}{m})\sigma^2 \;\leq\; \sigma^2.$$

Therefore, at the end of $M = \Theta(m)$ rounds, for each $j = 1, 2, \ldots, m$, all nodes have the average of the estimates in the path $\mathcal{P}_j$ that is perturbed by Gaussian noise with variance at most σ^2. Since $m = \Theta(D_n)$, we have $M = \Theta(D_n)$.

- At the end of the inner phase τ, nodes that were involved in a path use their estimate of the average along the path to update $\theta(\tau)$, while estimate of the nodes that were not involved in any route remain the same. A given node s_{ij} on a path updates its estimate via

$$\theta_{s_{ij}}(\tau+1) = \big\{1-\epsilon'(\tau)\big\}\theta_{s_{ij}}(\tau) + \epsilon'(\tau)\gamma_{ij}(M), \tag{5}$$

where $\epsilon'(\tau) = \mathcal{O}\left(\frac{1}{\tau+1/\delta}\right)$. On the other hand, using $\langle\cdot\,,\,\cdot\rangle$ to denote the Euclidean inner product, we have $\gamma_{ij}(M) = \langle w\,,\,\theta(\tau)\rangle + v_{s_{ij}}$, where w is the averaging vector of the route $\mathcal{P}_j$ with the entries $w(s_{\ell j}) = \frac{1}{m}$ for $\ell = 1, 2, \cdots, m$, and zero otherwise. Combining the scalar updates (5) yields the matrix-form update

$$\theta(\tau+1) = \theta(\tau) - \epsilon'(\tau)\big\{(I - W(\tau))\theta(\tau) + v'(\tau)\big\}, \tag{6}$$

where the matrix $W(\tau) = W(\tau; \mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m, \zeta)$ is a random averaging matrix induced by the choice of routes $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m$ and the random directions ζ. The noise vector $v'(\tau) \sim \mathcal{N}(0, C')$ is additive noise. Note that for any given time, the noise at different nodes are correlated via the matrix C', but for different time instants $\tau \neq \tau'$, the noise vectors $v'(\tau)$ and $v'(\tau')$ are independent. Moreover, from our earlier arguments, we have the upper bound $\max\limits_{i=1,\ldots,n} C'_{ii} \leq \sigma^2$.

IV. Proof of Theorem 1

We now turn to the proof of Theorem 1. At a high-level, the structure of the argument consists of decomposing the vector $\theta(\tau) \in \mathbb{R}^n$ into a sum of two terms: a component within the consensus subspace (meaning all values of the vector are identical), and a component in the orthogonal complement. Using this decomposition, the mean-squared error splits into a sum of two terms and we use standard techniques to bound them. As will be shown, these bounds depend on the parameter δ, noise variance, the initial MSE,

and finally the (inverse) spectral gap of the update matrix. The final step is to lower bound the spectral gap of our update matrix.

A. Setting up the proof

Recalling the averaging matrix $W(\tau)$ from the update (6), we define the Laplacian matrix $S(\tau) := I - W(\tau)$. We then define the average matrix $\overline{W} := \mathbb{E}[W(\tau)]$, where the expectation is taken place over the randomness due to the choice of routes;[3] in a similar way, we define the associated (average) Laplacian $\overline{S} := I - \overline{W}$. Finally, we define the rescaled quantities

$$\epsilon(\tau) := \lambda_2(\overline{S})\,\epsilon'(\tau), \quad L(\tau) := \frac{1}{\lambda_2(\overline{S})}\,S(\tau), \quad \text{and} \quad v(\tau) := \frac{1}{\lambda_2(\overline{S})}\,v'(\tau), \tag{7}$$

where we recall that $\lambda_2(\cdot)$ denotes the second smallest eigenvalue of a symmetric matrix. In terms of these rescaled quantities, our algorithm has the form

$$\theta(\tau+1) = \theta(\tau) - \epsilon(\tau)[L(\tau)\theta(\tau) + v(\tau)], \tag{8}$$

as stated previously in the update equation (4). Moreover, by construction, we have $v(\tau) \sim \mathcal{N}(0, C)$ where $C = \frac{1}{(\lambda_2(S))^2}C'$. We also, for theoretical convenience, set

$$\epsilon'(\tau) = \frac{1}{\lambda_2(\bar{S})(\tau + \frac{1}{\delta})}, \tag{9}$$

or equivalently $\epsilon(\tau) = \frac{1}{(\tau+\frac{1}{\delta})}$ for $\tau = 1, 2, \cdots$.

We first claim that the matrix $\overline{W}$ is symmetric and (doubly) stochastic. The symmetry follows from the fact that different routes do not collide, whereas the matrix is stochastic because every row of W (depending on whether the node corresponding to that row participates in a route or not) either represents an averaging along a route or is the corresponding row of the identity matrix. Consequently, we can interpret $\overline{W}$ as the transition matrix of a reversible Markov chain. It is an irreducible Markov chain, because within any updating round, there is a positive chance of averaging nodes that are in the same column or row, which implies that the associated Markov chain can transition from one state to any other in at most two steps. Moreover, the stationary distribution of the chain is uniform (i.e., $\pi = \vec{1}/n$).

We now use these properties to simplify our study of the sequence $\{\theta(\tau)\}_{\tau=1}^{\infty}$ generated by the update equation (8). Since $\overline{S}$ is real and symmetric, it has the eigenvalue decomposition $\overline{S} = U\Lambda U^T$, where $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}$ is a unitary matrix (that is, $U^T U = I_n$). Moreover, we have $\Lambda =$

[3] For the single cycle graph, there is only one route that involves all the nodes at each round, so $W(\tau)$ is deterministic in this case.

$\mathrm{diag}\{\lambda_1(\bar{S}), \lambda_2(\bar{S}), \cdots, \lambda_n(\bar{S})\}$, where $\lambda_i(\bar{S})$ is the eigenvalue corresponding to the eigenvector u_i, for $i = 1, \ldots, n$. Since $\bar{L} = \frac{1}{\lambda_2(S)}(I - \overline{W})$, the eigenvalues of $\bar{L}$ and $\overline{W}$ are related via

$$\begin{aligned}\lambda_i(\bar{L}) &= \frac{1}{\lambda_2(\bar{S})}(1 - \lambda_{n+1-i}(\overline{W})) \\ &= \frac{1}{1 - \lambda_{n-1}(\overline{W})}(1 - \lambda_{n+1-i}(\overline{W})).\end{aligned}$$

Since the largest eigenvalue of an irreducible Markov chain is one (with multiplicity one) [16], we have $1 = \lambda_n(\overline{W}) > \lambda_{n-1}(\overline{W}) \geq \cdots \geq \lambda_1(\overline{W})$, or equivalently

$$0 = \lambda_1(\bar{L}) < \lambda_2(\bar{L}) \leq \cdots \leq \lambda_n(\bar{L}),$$

with $\lambda_2(\bar{L}) = 1$. Moreover, we have $\bar{S}\vec{1} = \bar{L}\vec{1} = \vec{0}$, so that the first eigenvector $u_1 = \vec{1}/\sqrt{n}$ corresponds to the eigenvalue $\lambda_1(\bar{L}) = 0$. Let $\widetilde{U}$ denote the matrix obtained from U by deleting its first column, u_1. Since the smallest eigenvalue of $\bar{L}$ is zero, we may write $\bar{L} = \widetilde{U}\widetilde{\Lambda}\widetilde{U}^T$, where $\widetilde{\Lambda} = \mathrm{diag}\{\lambda_2(\bar{L}), \cdots \lambda_n(\bar{L})\}$, $\widetilde{U}^T\widetilde{U} = I_{n-1}$, and $\widetilde{U}\widetilde{U}^T = I_n - \frac{\vec{1}\vec{1}^T}{n}$. With this notation, our analysis is based on the decomposition

$$\theta(\tau) = \alpha(\tau)\frac{\vec{1}}{\sqrt{n}} + \widetilde{U}\beta(\tau), \tag{10}$$

where we have defined $\alpha(\tau) := \langle \vec{1}/\sqrt{n}\,, \theta(\tau) \rangle \in \mathbb{R}$ and $\beta(\tau) := \widetilde{U}^T\theta(\tau) \in \mathbb{R}^{n-1}$. Since $\vec{1}^T L(\tau) = \vec{0}^T$ for all $\tau = 1, 2, \cdots$, from the decomposition (10) and the form of the updates (8), we have the following recursions,

$$\alpha(\tau + 1) = \alpha(\tau) - \epsilon(\tau)\frac{\vec{1}^T}{\sqrt{n}}v(\tau), \quad \text{and} \tag{11}$$

$$\beta(\tau + 1) = \beta(\tau) - \epsilon(\tau)\big(\underline{L}(\tau)\beta(\tau) + \widetilde{U}^T v(\tau)\big). \tag{12}$$

Here $\underline{L}$ is an $(n-1) \times (n-1)$ matrix defined by the relation

$$U^T L(\tau) U = \begin{bmatrix} 0 & \vec{0}^T \\ \vec{0} & \underline{L}(\tau) \end{bmatrix}_{n \times n}.$$

B. Main steps

As we show, part (a) of the theorem requires some intermediate results of the proof of part (b). Accordingly, we defer it to the end of the section. With this set-up, we now state the two main technical lemmas that form the core of Theorem 1. Our first lemma concerns the behavior of the component sequences $\{\alpha(\tau)\}_{\tau=0}^{\infty}$ and $\{\beta(\tau)\}_{\tau=0}^{\infty}$ which evolve according to equations (11) and (12) respectively.

Lemma 2. *Given the random sequence $\{\theta(\tau)\}$ generated by the update equation* (4)*, we have*

$$\mathrm{MSE}(\theta(\tau)) = \underbrace{\frac{1}{n}\,\mathrm{var}\,(\alpha(\tau))}_{e_1(\tau)} + \underbrace{\frac{1}{n}\,\mathbb{E}[\|\beta(\tau)\|_2^2]}_{e_2(\tau)}. \tag{13}$$

Furthermore, $e_1(\tau)$ and $e_2(\tau)$ satisfy the following bounds:

(a) For each iteration $\tau = 1, 2, \ldots$, we have

$$e_1(\tau) \leq \frac{\sigma^2\,\delta}{[\lambda_2(\bar{S})]^2}. \tag{14}$$

(b) Moreover, for each iteration $\tau = 1, 2, \ldots$ we have

$$e_2(\tau) \;\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2}\,\frac{\log(\tau + \frac{1}{\delta} - 1)}{\tau + \frac{1}{\delta} - 1} \;+\; e_2(0)\,\frac{\frac{1}{\delta} - 1}{\tau + \frac{1}{\delta} - 1}, \tag{15}$$

From Lemma 2, we conclude that in order to guarantee an $\mathcal{O}(\frac{\sigma^2\delta}{[\lambda_2(\bar{S})]^2})$ bound on the MSE, it suffices to take τ such that

$$\frac{\frac{1}{\delta} - 1}{\tau + \frac{1}{\delta} - 1} \;\leq\; \frac{\sigma^2\,\delta}{e_2(0)[\lambda_2(\bar{S})]^2}, \quad \text{and} \quad \frac{\log(\tau + \frac{1}{\delta} - 1)}{\tau + \frac{1}{\delta} - 1} \;\leq\; \delta.$$

Note that the first inequality is satisfied when $\tau \geq \frac{e_2(0)}{\sigma^2\delta^2}\,[\lambda_2(\bar{S})]^2$. Moreover, doing a little bit of algebra, one can see that $\tau = \frac{2}{\delta}\log\frac{1}{\delta} - (\frac{1}{\delta} - 1)$ is sufficient to satisfy the second inequality. Accordingly, we take

$$\tau = \max\left\{\frac{2}{\delta}\log\frac{1}{\delta}\,,\,\frac{e_2(0)[\lambda_2(\bar{S})]^2}{\sigma^2\delta^2}\right\}$$

outer iterations.

The last part of the proof is to bound the second smallest eigenvalue of the Laplacian matrix $\bar{S}$. The following lemma, which we prove in Section IV-D to follow, addresses this issue. Recall that $\lambda_2(\cdot)$ denotes the second smallest eigenvalue of a matrix.

Lemma 3. *The averaged matrix $\bar{S}$ that arises from our protocol has the following properties:*

(a) For a cycle and a regular grid we have $\lambda_2(\bar{S}) = \Omega(1)$, and

(b) for a random geometric graph, we have $\lambda_2(\bar{S}) = \Omega(\frac{1}{\log n})$, with high probability.

It is important to note that the averaged matrix $\bar{S}$ is *not the same* as the graph Laplacian that would arise from standard averaging on these graphs. Rather, as a consequence of establishing many paths and averaging along them in each inner phase, our protocol ensures that the matrix behaves essentially like the graph Laplacian for the fully connected graph.

As established previously, each outer step requires $M = \mathcal{O}(D_n)$ iterations. Therefore, we have shown

that it is sufficient to take a total of

$$T \;=\; \mathcal{O}\left(D_n \;\max\left\{\frac{2}{\delta}\log\frac{1}{\delta}\,,\;\frac{e_2(0)[\lambda_2(\bar{S})]^2}{\sigma^2\delta^2}\right\}\right)$$

transmissions per edge in order to guarantee a $\frac{3\,\sigma^2\delta}{[\lambda_2(\bar{S})]^2}$ bound on the MSE. As we will see in the next section, assuming that the initial values are fixed, we have $e_1(0) = 0$, hence $\mathrm{MSE}(\theta(0)) = e_2(0)$. The claims in Theorem 1 then follow by standard calculations of the diameters of the various graphs and the result of the Lemma 3.

It remains to prove the two technical results, Lemma 2 and 3, and we do so in the following sections.

C. Proof of Lemma 2

We begin by observing that

$$\mathbb{E}\left[(\theta(\tau)-\bar{\theta}\vec{1})(\theta(\tau)-\bar{\theta}\vec{1})^T\right] = F_1 + F_2 + F_3,$$

where $F_1 := \mathbb{E}\left[(\alpha(\tau)-\sqrt{n}\bar{\theta})^2\right]\frac{\vec{1}\vec{1}^T}{n}$, the second term is given by $F_2 := \mathbb{E}\left[\widetilde{U}\beta(\tau)\beta(\tau)^T\widetilde{U}^T\right]$, and

$$F_3 := \mathbb{E}\left[(\alpha(\tau)-\sqrt{n}\bar{\theta})\,\frac{\vec{1}}{\sqrt{n}}\,\beta(\tau)^T\widetilde{U}^T\right] \;+\; \mathbb{E}\left[(\alpha(\tau)-\sqrt{n}\bar{\theta})\,\widetilde{U}\beta(\tau)\,\frac{\vec{1}^T}{\sqrt{n}}\right].$$

Since $\widetilde{U}$ has orthonormal columns, all orthogonal to the all one vector ($\vec{1}^T\widetilde{U} = \vec{0}$), it follows that $\mathrm{trace}(F_2) = \mathbb{E}\big[\|\beta(\tau)\|_2^2\big]$, and $\mathrm{trace}(F_3) = 0$.

It remains to compute $\mathrm{trace}(F_1)$. Unwrapping the recursion (11) and using the fact that initialization $\theta(0)$ implies $\alpha(0) = \sqrt{n}\bar{\theta}$ yields

$$\alpha(\tau) = \sqrt{n}\bar{\theta} - \sum_{l=0}^{\tau-1}\epsilon(l)\,\langle\frac{\vec{1}}{\sqrt{n}}\,,\,v(l)\rangle, \tag{16}$$

for all $\tau = 1, 2, \ldots$. Since $v(l)$, $l = 0, 1, \cdots, \tau-1$, are zero mean random vectors, from equation (16) we conclude that $\mathbb{E}[\alpha(\tau)] = \sqrt{n}\bar{\theta}$ [4] and accordingly, $\mathrm{trace}(F_1) = \mathrm{var}\,(\alpha(\tau))$. Recalling the definition of the MSE (1) and combining the pieces yields the claim (13).

(a) From equation (16), it is clear that each $\alpha(\tau)$ is Gaussian with mean $\sqrt{n}\bar{\theta}$. It remains to bound the

[4]Here we have assumed that the initial values, $\theta_i(0)$ $i = 1, 2, \cdots, n$, are given (fixed).

variance. Using the i.i.d. nature of the sequence $v(i) \sim \mathcal{N}(0, C)$, we have

$$\begin{aligned}\mathrm{var}\,(\alpha(\tau)) &= \mathbb{E}\Big[\big(\sum_{l=0}^{\tau-1}\epsilon(l)\langle\frac{\vec{1}}{\sqrt{n}}\,,\,v(l)\rangle\big)^2\Big] \\ &= \sum_{l=0}^{\tau-1}\frac{\epsilon(l)^2}{n}\langle\vec{1}\,,\,C\vec{1}\rangle \\ &= \sum_{l=0}^{\tau-1}\epsilon'(l)^2\frac{\langle\vec{1}\,,\,C'\vec{1}\rangle}{n},\end{aligned}$$

where we have recalled the rescaled quantities (7). Recalling the fact that $C'_{ii} \leq \sigma^2$ and using the Cauchy-Schwarz inequality, we have $C'_{ij} \leq \sqrt{C'_{ii}C'_{jj}} \;\leq\; \sigma^2$. Hence, we obtain

$$\begin{aligned}\mathrm{var}\,(\alpha(\tau)) &\leq n\sigma^2\;\sum_{l=0}^{\tau-1}\epsilon'(l)^2 \\ &= \frac{n\sigma^2}{[\lambda_2(\bar{S})]^2}\sum_{l=0}^{\tau-1}\frac{1}{(\frac{1}{\delta}+l)^2} \\ &\leq \frac{n\sigma^2}{[\lambda_2(\bar{S})]^2}\int_{\frac{1}{\delta}}^{\infty}\frac{1}{x^2}dx \;=\frac{n\,\sigma^2\delta}{[\lambda_2(\bar{S})]^2};\end{aligned}$$

from which rescaling by $1/n$ establishes the bound (14).

(b) Defining $H(\beta(\tau), v(\tau)) = \underline{L}(\tau)\beta(\tau) + \widetilde{U}^T v(\tau)$, the update equation (12) can be written as

$$\beta(\tau+1) \;=\; \beta(\tau) \;-\; \epsilon\,(\tau)H(\beta(\tau), v(\tau)),$$

for $\tau = 1, 2, \cdots$. In order to upper bound $e_2(\tau+1)$, defined in (13), we need to control $e_2(\tau+1) - e_2(\tau)$. Doing some algebra yields

$$\begin{aligned}e_2(\tau+1) - e_2(\tau) \;&=\; \frac{1}{n}\,\mathbb{E}\left[\langle\beta(\tau+1) - \beta(\tau)\,,\,\beta(\tau+1) + \beta(\tau)\rangle\right] \\ &=\; \frac{1}{n}\,\mathbb{E}\left[\langle-\epsilon\,(\tau)H(\beta(\tau, v(\tau)))\,,\,-\epsilon\,(\tau)H(\beta(\tau, v(\tau))) + 2\beta(\tau)\rangle\right],\end{aligned}$$

and hence

$$e_2(\tau+1) - e_2(\tau) \;=\; \frac{1}{n}\,\epsilon(\tau)^2\,\mathbb{E}\left[\|H(\beta(\tau), v(\tau))\|_2^2\right] \;-\; \frac{2\epsilon(\tau)}{n}\,\mathbb{E}\left[\langle H(\beta(\tau), v(\tau))\,,\,\beta(\tau)\rangle\right].$$

Since $\beta(\tau)$ is independent of both $L(\tau)$ and $v(\tau)$, by conditioning on the $\beta(\tau)$ and using the tower property of expectation, we obtain

$$\mathbb{E}\left[\langle H(\beta(\tau), v(\tau))\,,\,\beta(\tau)\rangle\right] \;=\; \mathbb{E}\left[\langle\mathbb{E}\left[\underline{L}\right]\beta(\tau)\,,\,\beta(\tau)\rangle\right].$$

By construction all the eigenvalues of $\mathbb{E}\left[\underline{L}\right]$ are greater than one, hence

$$\langle \mathbb{E}\left[\underline{L}\right]\beta(\tau)\,,\,\beta(\tau)\rangle \;\geq\; \|\beta(\tau)\|_2^2.$$

Putting the pieces together, we obtain

$$\begin{aligned} e_2(\tau+1) \;&\leq\; \frac{1}{n}\,\epsilon(\tau)^2\,\mathbb{E}\left[\|H(\beta(\tau),v(\tau))\|_2^2\right] \;+\; (1-2\epsilon(\tau))\,e_2(\tau) \\ &=\; \frac{1}{n}\,\epsilon(\tau)^2\,\underbrace{\mathbb{E}\left[\|\underline{L}(\tau)\beta(\tau)\|_2^2\right]}_{F_1} + \frac{1}{n}\,\epsilon(\tau)^2\,\underbrace{\mathbb{E}\left[\|\widetilde{U}^T v(\tau)\|_2^2\right]}_{F_2} + (1-2\epsilon(\tau))\,e_2(\tau), \end{aligned} \tag{17}$$

where we used the fact that $\mathbb{E}\left[\langle \underline{L}(\tau)\beta(\tau)\,,\,\widetilde{U}^T v(\tau)\rangle\right] = 0$. We continue by upper bounding the terms $F_1 = \mathbb{E}\left[\|\underline{L}(\tau)\beta(\tau)\|_2^2\right]$, and $F_2 \;=\; \mathbb{E}\left[\|\widetilde{U}^T v(\tau)\|_2^2\right]$. First, we bound the former. By definition of the l_2-operator norm, we have

$$\mathbb{E}\left[\|\underline{L}(\tau)\beta(\tau)\|_2^2\right] \;\leq\; \mathbb{E}\left[\|\!|\underline{L}(\tau)|\!\|_2^2\,\|\beta(\tau)\|_2^2\right].$$

On the other hand, using the fact that $\underline{L}(\tau) = \frac{1}{\lambda_2(\bar{S})}\,\widetilde{U}^T\,(I - W(\tau))\,\widetilde{U}$ (recall the identities of the Section IV-A) yields[5]

$$|\!|\underline{L}(\tau)|\!\|_2 \;\leq\; \frac{1}{\lambda_2(\bar{S})}\,(1 + |\!|W(\tau)|\!\|_2) \;=\; \frac{2}{\lambda_2(\bar{S})}.$$

Therefore, we have the following bound on F_1

$$F_1 \;\leq\; \frac{4}{[\lambda_2(\bar{S})]^2}\,\mathbb{E}\left[\|\beta(\tau)\|_2^2\right]. \tag{18}$$

Turning to term F_2, we have

$$F_2 \;=\; \mathbb{E}\left[v(\tau)^T(I - \frac{\vec{1}\vec{1}^T}{n})v(\tau)\right] \;\leq\; \mathrm{trace}\left(\mathrm{cov}(v(\tau))\right) \;\leq\; \frac{n\sigma^2}{[\lambda_2(\bar{S})]^2}. \tag{19}$$

Substituting the inequalities (18) and (19) into (17), we obtain the following recursive bound on $e_2(\tau+1)$

$$e_2(\tau+1) \;\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2}\,\epsilon(\tau)^2 \;+\; \left(1 - 2\epsilon(\tau) + \frac{4\epsilon(\tau)^2}{[\lambda_2(\bar{S})]^2}\right)\,e_2(\tau).$$

[5]Let v be an eigenvector of the matrix $W(\tau)$ corresponding to the eigenvalue $\lambda \neq 1$. Since $\vec{1}^T v = 0$, there exist an $(n-1)$-dimensional vector u such that $v = \widetilde{U}u$. Therefore we have,

$$\widetilde{U}^T(I - W(\tau))\widetilde{U}u \;=\; \widetilde{U}^T(I - W(\tau))v \;=\; (1-\lambda)\widetilde{U}^T v \;=\; (1-\lambda)u.$$

So by subtracting one from the eigenvalues of $\widetilde{U}^T(I - W(\tau))\widetilde{U}$, we obtain the non-one eigenvalues of $W(\tau)$.

Recall the definitions (7) and (9). If $\delta \leq \frac{[\lambda_2(\bar{S})]^2}{4}$, then $1 - 2\epsilon(\tau) + \frac{4\epsilon(\tau)^2}{[\lambda_2(\bar{S})]^2} \leq 1 - \epsilon(\tau)$, and hence we have

$$e_2(\tau+1) \;\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2}\epsilon(\tau)^2 \;+\; (1-\epsilon(\tau))e_2(\tau), \tag{20}$$

for all $\tau = 1, 2, \cdots$. Unwrapping the inequality (20) yields

$$e_2(\tau+1) \;\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2} \sum_{k=0}^{\tau} \epsilon(k)^2 \prod_{l=k+1}^{\tau} (1-\epsilon(l)) \;+\; \prod_{l=0}^{\tau}(1-\epsilon(l))\, e_2(0). \tag{21}$$

On the other hand, the product $\prod_{l=k+1}^{\tau}(1-\epsilon(l))$ forms a telescopic series and is equal to $\frac{k+\frac{1}{\delta}}{\tau+\frac{1}{\delta}}$. Substituting this fact into the equation (21) yields

$$\begin{aligned} e_2(\tau+1) \;&\leq\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2} \sum_{k=0}^{\tau} \frac{1}{(k+\frac{1}{\delta})\,(\tau+\frac{1}{\delta})} \;+\; e_2(0)\,\frac{\frac{1}{\delta}-1}{\tau+\frac{1}{\delta}} \\ &\overset{(a)}{\leq}\; \frac{\sigma^2}{[\lambda_2(\bar{S})]^2}\,\frac{\log(\tau+\frac{1}{\delta})}{\tau+\frac{1}{\delta}} \;+\; e_2(0)\,\frac{\frac{1}{\delta}-1}{\tau+\frac{1}{\delta}}, \end{aligned}$$

where step (a) uses the following inequality

$$\sum_{k=0}^{\tau} \frac{1}{k+\frac{1}{\delta}} \;\leq\; \int_{\frac{1}{\delta}-1}^{\tau+\frac{1}{\delta}} \frac{1}{x}dx \;\leq\; \log(\tau+\frac{1}{\delta}),$$

valid for $\delta \in (0, \frac{1}{2})$.

D. Proof of Lemma 3

In the case of cycle there is only one averaging path and all the nodes are involved in that at each round so the averaging matrix, W, is fixed. More precisely, we have $\overline{W} = W = \frac{1}{n}\vec{1}\vec{1}^T$. Therefore, $\overline{W}$ is a rank 1 matrix with $\lambda_{n-1}(\overline{W}) = 0$ and accordingly we have $\lambda_2(\overline{S}) = 1 - \lambda_{n-1}(\overline{W}) = 1$.

For the case of grid or random geometric graphs, we use the Poincare inequality [11]. A version of this theorem can be stated as follows: Let $A = [a_{ij}]$ denote the transition matrix of an irreducible aperiodic time reversible Markov chain with stationary distribution π. For each ordered pair of nodes (s, u) in the transition diagram, choose one and only one path $\eta_{su} = (s, s_1, s_2, \cdots, s_l, u)$ between s and u and define

$$|\eta_{su}| := \frac{1}{\pi(s)a_{ss_1}} + \frac{1}{\pi(s_1)a_{s_1s_2}} + \cdots + \frac{1}{\pi(s_l)a_{s_lu}}. \tag{22}$$

Then the Poincare coefficient is

$$\kappa := \max_{e \in E'} \sum_{\eta_{su} \ni e} |\eta_{su}|\pi(s)\pi(u), \tag{23}$$

where E' is the set of directed edges formed in the previous step. Defining this quantity, the theorem states

that $\lambda_{n-1}(A) \leq 1 - \frac{1}{\kappa}$ or equivalently,

$$1 - \lambda_{n-1}(A) \geq \frac{1}{\kappa}. \tag{24}$$

We apply this theorem to the Markov chain formed by $\overline{W}$; the idea is to upper bound its Poincare coefficient.

1) Grid: We first define a path η_{su} for every pair of nodes $\{s, u\}$. Two different cases can be distinguished here. For an illustration of the path η_{su} see Figure 4.

a) Case 1: Nodes s and u do not belong to the same column or row. In this case, we consider a two-hop path $\eta_{su} = (s \to w \to u)$, where $w = (x_u, y_s)$ is the vertex of the rectangle constructed by s and u. Note that x_u is the x-coordinate of u and y_s is the y-coordinate of s. Since nodes $\{s,\ w\}$ and $\{w,\ u\}$ are averaged $\frac{1}{2}$ of the time, we have $\overline{W}_{sw} = \overline{W}_{wu} = \frac{1}{2m}$. Substituting this into (22) and using the fact that $\pi = \frac{1}{n}\vec{1}$ yields

$$|\eta_{su}| = \frac{1}{\overline{W}_{sw}\pi(s)} + \frac{1}{\overline{W}_{wu}\pi(w)} = 4mn.$$

b) Case 2: Nodes s and u belong to the same row or column. In this case, we set $\eta_{su} = (s \to u)$ which leads to

$$|\eta_{su}| = \frac{1}{\overline{W}_{su}\pi(s)} = 2mn.$$

Moreover, a given edge $e = (s \to w)$ is involved in at most m paths. As node u varies in the corresponding column or row, we obtain $m - 1$ paths in case 1, and one path in case 2.

Combining the pieces, we compute the Poincare coefficient

$$\kappa = \max_{e \in E'} \sum_{\eta_{su} \ni e} |\eta_{su}|\pi(s)\pi(u) \leq m\frac{4mn}{n^2} = 4.$$

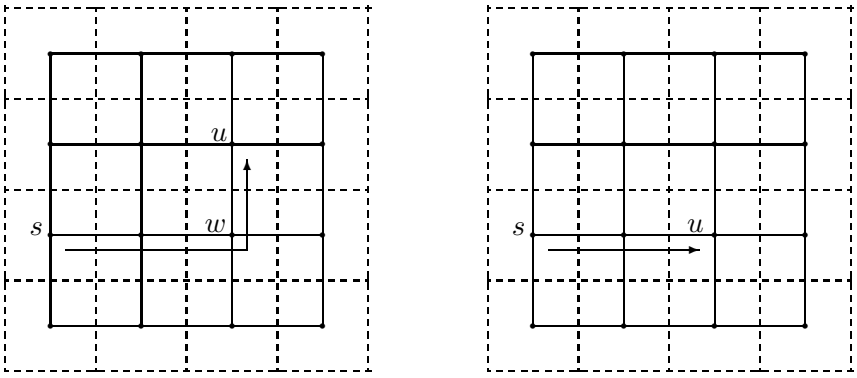
Finally, from equation (24), we have

$$\lambda_2(\bar{S}) = 1 - \lambda_{n-1}(\overline{W}) \geq \frac{1}{\kappa} \geq \frac{1}{4}$$

which concludes the proof for the case of a grid-structured graph.

2) Random geometric graph: For the RGG, we follow the same proof structure: namely, we first find a path for each pair of nodes $\{s, u\}$, and then upper bound the Poincare coefficient for the Markov chain $\overline{W}$. We first introduce some useful notation. Let $\mathcal{C} : \mathcal{V} \to \{1, 2, \cdots, m\}^2$ be the mapping that takes a node as its input and returns the sub-square of that node. More precisely, for some $s \in \mathcal{V}$ we have

$$\mathcal{C}(s) = (i, j) \quad \text{if } s \in (i, j)\text{-th square } i, j = 1, 2, \cdots, m.$$



(a) Case 1. (b) Case 2.

Fig. 4. Illustration of the path η_{su} for a grid-structured graph. (a) Case 1, where nodes s and u do not belong to the same column or row. (b) Case 2, where nodes s and u belong to the same column or row. This choice of η_{su} yield a tight upper bound on the Poincare coefficient.

Furthermore, we enumerate the nodes in square $\mathcal{C}(s) = (i, j)$ from 1 to n_{ij} where n_{ij} denotes the total number of nodes in $\mathcal{C}(s)$. We refer to the label of node s as $\mathcal{N}_{\mathcal{C}(s)}(s)$ where $\mathcal{N}_{\mathcal{C}(s)}(.)$ is the enumeration operator for the square $\mathcal{C}(s)$. Also let $n^* = \min_{i,j} n_{ij}$ denote the minimum number of nodes in one sub-square which by assumption is greater than $a \log n$ for some constant a. We split the problem into three different cases. Figure 5 illustrates these there different cases.

a) Case 1: Nodes s and u do not belong to the the same column or row. In this case, a two hop path $\eta_{su} = (s \to w \to u)$ is considered. First, we pick $\mathcal{C}(w)$, the vertex of the rectangle constructed by $\mathcal{C}(s)$ and $\mathcal{C}(u)$ with the same x-coordinate as $\mathcal{C}(u)$ and the same y-coordinate as $\mathcal{C}(s)$. Now choose a node, w, inside $\mathcal{C}(w)$ such that

$$\mathcal{N}_{\mathcal{C}(w)}(w) = \mathcal{N}_{\mathcal{C}(s)}(s) + \mathcal{N}_{\mathcal{C}(u)}(u) \mod n^*. \tag{25}$$

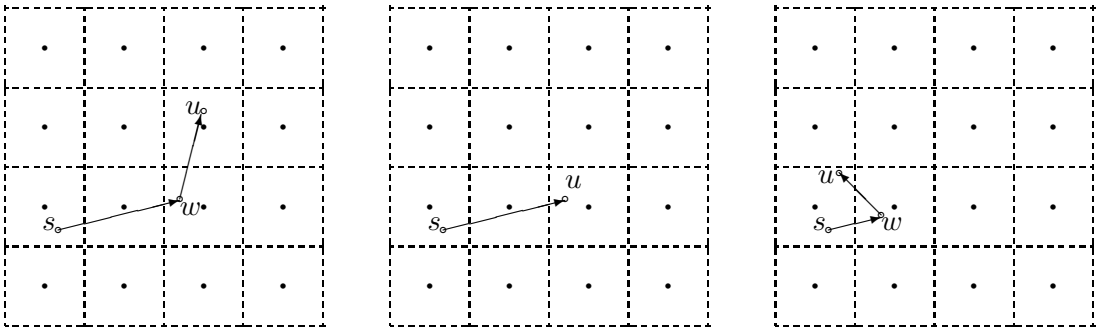
Since each square has at least n^* nodes, such a choice can be made. On the other hand, since nodes in each square is picked uniformly at random in the averaging phase and there are at most $b \log n$ nodes in each square (for some constant b) we have $\overline{W}_{sw}, \overline{W}_{wu} \geq \frac{1}{2m(b \log n)^2}$, where the factor of 2 is due to the choice of ζ, the averaging direction. Substituting this inequality into (22), we obtain

$$|\eta_{su}| = \frac{1}{\overline{W}_{sw}\pi(s)} + \frac{1}{\overline{W}_{wu}\pi(w)} \ \leq \ 4b^2 mn \, (\log n)^2.$$

Furthermore, from equation (25), we see that for a fixed s there are at most $\frac{b}{a}$ nodes in the square $\mathcal{C}(u)$ that result in choosing w. Therefore, edge $e : (s \to w)$ is involved in at most $\frac{b}{a}(m-1)$ such paths.

b) Case 2: Nodes s and u belong to the same row or column. In this case, by setting $\eta_{su} = (s \to u)$, we obtain

$$|\eta_{su}| = \frac{1}{\overline{W}_{su}\pi(s)} \leq 2b^2 mn (\log n)^2.$$



(a) Case 1. (b) Case 2. (c) Case 3.

Fig. 5. Illustration of the path η_{su} for the case of RGG. (a) Case 1, where nodes s and u belong to the sub-squares in different row and columns (b) Case 2, where nodes s and u belong to the sub-squares in the same row or column. (c) Case 3, nodes s and u belong to the same square.

Note that there is only one path containing e of this type.

c) Case 3: Nodes s and u belong to the same square, meaning $\mathcal{C}(s) = \mathcal{C}(u)$. In this case a node w is chosen in a square adjacent to $\mathcal{C}(s)$ according to (25) such that $\mathcal{C}(w)$ is to the right of $\mathcal{C}(s)$; unless $\mathcal{C}(s)$ is in the last column, in which case $\mathcal{C}(w)$ is to the left of $\mathcal{C}(s)$. The same argument as case 1 would give us a bound on $|\eta_{su}|$. As for the upper bound on the number of paths: the edge $e : (s \to w)$ is involved in at most $\frac{b}{a}$ such paths.

Combining all the pieces, we obtain

$$|\eta_{su}| \;\leq\; 4b^2 mn(\log n)^2 \quad \forall\, s, u \in \mathcal{V},$$

and

$$\max_{e \in E'} \sum_{s,u} \mathbb{I}\left\{\eta_{su} \ni e\right\} \;\leq\; m\,\frac{b}{a} + 1.$$

Substituting these two inequalities into (23) yields

$$\begin{aligned} \kappa &\leq \left(m\,\frac{b}{a} + 1\right) \frac{4b^2\, mn\,(\log n)^2}{n^2} \\ &\leq \frac{2mb}{a}\,\frac{4b^2\, mn\,(\log n)^2}{n^2} \\ &= c_1 \log n \end{aligned}$$

for some constant c_1. Therefore, from Poincare Theorem, we have

$$\lambda_2(\bar{S}) = 1 - \lambda_{n-1}(\overline{W}) \geq \frac{1}{\kappa} \geq \frac{1}{c_1 \log n}$$

which concludes the second part of Lemma 3.

E. Proof of part (a) of Theorem 1

We now return to the proof of part (a) of Theorem 1. Combining equations (10) and (16) yields

$$\theta(\tau) \,=\, (\bar{\theta} \,-\, w(\tau))\,\vec{1} \,+\, \widetilde{U}\beta(\tau), \tag{26}$$

where $w(\tau) \,=\, \frac{1}{\sqrt{n}} \,\sum_{l=0}^{\tau-1} \epsilon(l)\,\langle \frac{\vec{1}}{\sqrt{n}}\,,\, v(l)\rangle$. As previously established, we know that $\mathbb{E}\,[w(\tau)] \,=\, 0$ and $\mathrm{var}\,(w(\tau)) \,\leq\, \frac{\sigma^2\delta}{[\lambda_2(S)]^2}$ for all $\tau \,=\, 1, 2, \cdots$. Therefore, invoking a result on convergence of series with bounded variance (Theorem 8.3 from Chapter 1 of [14]), we have

$$w(\tau) \xrightarrow{\text{a.s.}} w \quad \text{as } \tau \to \infty, \tag{27}$$

for some random variable w. Since $w(\tau)$ is a sum of independent Gaussian random variables (and hence Gaussian), it is absolutely integrable [14]. Therefore, we have $\mathbb{E}\,[w] \,=\, \lim_{\tau\to\infty} \mathbb{E}\,[w(\tau)] \,=\, 0$ and also $\mathrm{var}\,(w) = \lim_{\tau\to\infty} \mathrm{var}\,(w(\tau)) \leq \frac{\sigma^2\delta}{[\lambda_2(S)]^2}$.

Now we move on to the next part of the proof, analyzing the sequence $\{\beta(\tau)\}_{\tau=1}^{\infty}$ using techniques from stochastic approximation theory (e.g., see the books [21], [6]). These techniques apply to recursions that generate a state sequence $\{\theta(t)\}_{t=1}^{\infty}$ according to

$$\theta(t+1) \,=\, \theta(t) \,-\, \epsilon(t)\,H(\theta(t), v(t)) \quad t = 1, 2, \cdots,$$

where $v(t)$ is the noise vector that models the randomness coming into play in the algorithm. The parameter $\epsilon(t)$ is a positive step size, and the sequence $\{\epsilon(t)\}_{t=1}^{\infty}$ is required to satisfy the conditions $\sum_{t=1}^{\infty} \epsilon(t) = \infty$ and $\sum_{t=1}^{\infty} \epsilon(t)^{\alpha} < \infty$ for some $\alpha > 1$. The asymptotic behavior of these stochastic updates can be analyzed in terms of the ordinary differential equation (ODE)

$$\frac{d\gamma(\zeta)}{d\zeta} = -h(\gamma), \tag{28}$$

where $h(\theta) := \mathbb{E}[H(\theta, v)]$. Under mild regularity conditions, it is known that $\theta(t) \xrightarrow{\text{a.s.}} \gamma^*$, where γ^* is the attractor of the ODE (28).

Recalling the update equation (12), our problem can be cast within this framework. In particular, the state sequence is $\{\beta(\tau)\}_{\tau=1}^{\infty}$, the noise sequence is formed by zero-mean i.i.d. random vectors, the decreasing sequence is $\epsilon(\tau) = 1/(\frac{1}{\delta} + \tau)$, and finally $H(\beta, v) = (\underline{L}\beta + \widetilde{U}^T v)$ is a linear function with $h(\beta) = \mathbb{E}[\underline{L}]\beta$. Note because we removed the zero eigenvalue from the average Laplacian matrix, the matrix $\mathbb{E}[\underline{L}]$ has all positive eigenvalues, and so $\gamma^* \,=\, 0$ is the unique stable point of the linear differential equation

$\frac{d\gamma(\zeta)}{d\zeta} = -\mathbb{E}[\underline{L}]\gamma$. Therefore, an application of the ODE method [21], [6] guarantees that

$$\beta(\tau) \xrightarrow{a.s.} 0 \quad \text{as } \tau \to \infty. \tag{29}$$

Substituting the results (27) and (29) into equation (26), we obtain

$$\theta(\tau) \xrightarrow{\text{a.s.}} (\bar{\theta} - w)\vec{1} \quad \text{as } \tau \to \infty.$$

In other words, nodes will almost surely reach a consensus; moreover, the consensus value, $\widetilde{\theta} = \bar{\theta} - w$, is within $\frac{\sigma^2\delta}{[\lambda_2(\bar{S})]^2}$ distance of the true sample mean.

V. Simulation Results

In order to demonstrate the effectiveness of the proposed algorithm, we conducted a set of simulations. More specifically, we apply the proposed algorithm to four nearest-neighbor square grids of different sizes. We initially generate the data $\theta_i(0)$, $i = 1, 2, \cdots, n$ as random $N(1, 1)$ variables and fix them throughout the simulation. So for each run of the algorithm the initial data is fixed. In implementing the algorithm, we adopt $\sigma^2 = 1$ as the channel noise variance, and we set the tolerance parameter $\delta = 0.1$, leading to the step size $\epsilon(\tau) = \frac{1}{10+\tau}$. We estimated the mean-squared error, defined in equation (1), by taking the average over 50 sample paths. As discussed in Section III, every outer phase update requires $M = \mathcal{O}(\sqrt{n})$ time steps.

Figure 6 shows the mean-squared error versus the number of outer loop iterations; the panel contains two different curves, one for a graph with $n = 30^2$ nodes, and the other for $n = 50^2$ nodes. As expected, the MSE monotonically decreases as the number of iterations increases, showing convergence of the algorithm. More importantly, the gap between the two plots is negligible. This phenomenon, which is predicted by our theory, is explored further in our next set of experiments.

In order to study the network scaling of the grid more precisely, for a given set of graph sizes, we compute the number of the *outer iterations* $\tau = \tau(n, \delta)$, such that $\text{MSE}(\theta(\tau M)) \leq \sigma^2\delta$. Recall that this stopping time is the focus of Theorem 1(b). Figure 7 provides a box plot of this stopping time τ versus the graph size n. Theorem 1(b) predicts that this stopping time should be inversely proportional to the spectral gap of the Laplacian matrix $\overline{S}$, which for the grid scales as $\Omega(1)$ (in particular, see Lemma 3). As shown in Figure 7, over a range of graphs of size varying from $n = 1000$ to $n = 10000$, the stopping time is roughly constant ($\tau \approx 25$), which is consistent with the theory.

VI. Discussion

In this paper, we proposed and analyzed a two-phase graph-respecting algorithm for computing averages in a network, where communication is modeled as an additive white Gaussian noise channel. We showed

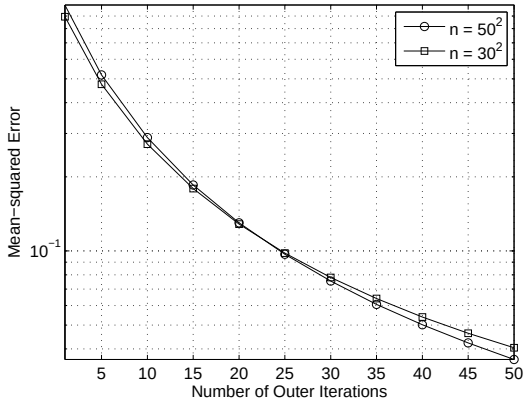


Fig. 6. Mean-squared error versus the number of outer loop iterations for grids with $n \in \{30^2, 50^2\}$ nodes. As expected the MSE monotonically decreases, which supports the convergence claim.

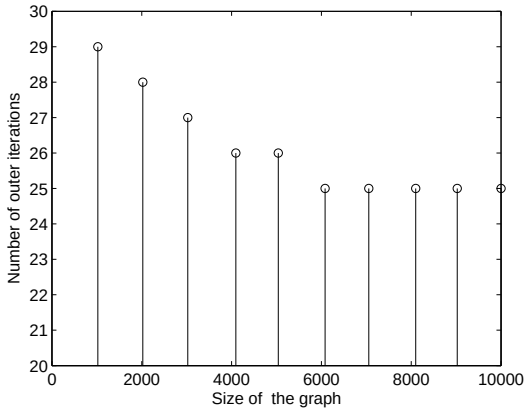


Fig. 7. Stopping time $\tau = \tau(n, \delta)$ vs. the graph size n. For different graph sizes, we compute the first outer phase time instance, $\tau(n, \delta)$, such that $\mathrm{MSE}(\theta(\tau M)) \leq \sigma^2 \delta$. Here we have fixed the parameters to $\sigma^2 = 1$, and $\delta = 0.1$. As you can see, over a range of graphs of size varying from 1000 to 10000, this stopping time is roughly constant (≈ 25), which is consistent with the theory (Theorem 1(b) and Lemma 3).

that it achieves consensus, and we characterized the rate of convergence as a function of the graph topology and graph size. For our algorithm, this network scaling is within logarithmic factors of the graph diameter, showing that it is near-optimal, since the graph diameter provides a lower bound for any algorithm.

There are various issues left open in this work. First, while the AWGN model is more realistic than noiseless communication, many channels in wireless networks may be more complicated, for instance involving fading, interference and other types of memory. In principle, our algorithm could be applied to such channels and networks, but its behavior and associated convergence rates remain to be analyzed. In a separate direction, it is also worth noting that gossip-type algorithms can be used to solve more complicated types of problems, such as distributed optimization problems (e.g., [25], [28], [13]). Studying the issue of near-optimal network scaling for such problems is also of interest.

Acknowledgements

NN and MJW were partially supported by NSF grant CCF-0545862 from the National Science Foundation, and AFOSR-09NL184 grant from the Air Force Office of Scientific Research.

REFERENCES

[1] O. Ayaso, D. Shah, and M. Dahleh. Information theoretic bounds for distributed computation over networks of point-to-point channels. In *International Symposium on Information Theory*, 2008.

[2] T. C. Aysal, M. J. Coates, and M. G. Rabbat. Distributed average consensus with dithered quantization. *IEEE Transactions on Signal Processing*, 56:4905–4918, 2008.

[3] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, 57:2748–2761, 2009.

[4] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Proc. IEEE International Symposium on Information Theory*, 2010.

[5] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli. Gossip along the way: order-optimal consensus through randomized path averaging. In *Forty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, Sep 2007.

[6] A. Benveniste, M. Metivier, and P. Priouret. *Stochastic approximations and adaptive algorithms*. Springer-Verlag, New York, 1990.

[7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52:2508–2530, 2006.

[8] F. Cattivelli and A. H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035–1048, March 2010.

[9] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[10] M. H. DeGroot. Reaching a consensus. *J. Amer. Stat. Assoc.*, 69:118–121, 1974.

[11] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Applied Probability*, 1:36–61, 1991.

[12] A. G. Dimakis, A. Sarwate, and M. J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Processing*, 53:1205–1216, March 2008.

[13] J. Duchi, A. Agawarl, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. Technical Report arXiv:1005.2012, UC Berkeley, May 2010.

[14] Rick Durrett. *Probability: Theory and Examples*. Thomson Learning, 2005.

[15] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM J. on Control and Optimization*, 2007. To appear.

[16] G.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, Clarendon Press, Oxford, 1992.

[17] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Trans. on Inf. Theory*, 46(2):388–404, Mar 2000.

[18] Y. Hatano, A. K. Das, and M. Mesbahi. Agreement in presence of noise: pseudogradients on random geometric networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, December 2005.

[19] S. Kar and J. M. F. Moura. Distributed consensus algorithm in sensor networks with imperfect communication: link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(5):355–369, Jan 2009.

[20] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. IEEE Conf. Foundation of Computer Science (FOCS)*, 2003.

[21] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, 2003.

[22] C. G. Lopes and A. H. Sayed. Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, 55(8):4064–4077, August 2007.

[23] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, July 2008.

[24] B. Nazer, A. G. Dimakis, and M. Gastpar. Neighborhood gossip: Concurrent averaging through local interference. In *Proc. IEEE ICASSP*, 2009.

[25] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54:48–61, 2009.

[26] M. Penrose. *Oxford studies in probability, Random Geometric Graphs*. Oxford Univ. Press, Oxford U.K., 2003.

[27] R. Rajagopal and M. J. Wainwright. Network-based consensus averaging with general noisy channels. *IEEE Transactions on Signal Processing*, Jan 2011.

[28] S. Sundhar Ram, A. Nedic, and V. V. Veeravalli. Distributed subgradient projection algorithm for convex optimization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3653–3656, 2009.

[29] H. I. Su and A. El Gamal. Distributed lossy averaging. In *Proc. IEEE International Symposium on Information Theory*, 2009.

[30] J. Tsitsiklis. *Problems in decentralized decision-making and computation*. PhD thesis, Department of EECS, MIT, 1984.

Nima Noorshams received his B.Sc. from Sharif University of Technology, Tehran, Iran, in 2007. He is currently pursuing his M.Sc. degree in the department of Statistics and his Ph.D. degree in the department of Electrical Engineering & Computer Science at University of California, Berkeley. His current research interests include stochastic approximation methods, graphical models, statistical signal processing, and modern coding theory.

Martin Wainwright is currently an associate professor at University of California at Berkeley, with a joint appointment between the Department of Statistics and the Department of Electrical Engineering and Computer Sciences. He received a Bachelor's degree in Mathematics from University of Waterloo, Canada, and Ph.D. degree in Electrical Engineering and Computer Science (EECS) from Massachusetts Institute of Technology (MIT). His research interests include coding and information theory, machine learning, mathematical statistics, and statistical signal processing. He has been awarded an Alfred P. Sloan Foundation Fellowship, an NSF CAREER Award, the George M. Sprowls Prize for his dissertation research (EECS department, MIT), a Natural Sciences and Engineering Research Council of Canada 1967 Fellowship, an IEEE Signal Processing Society Best Paper Award in 2008, and several outstanding conference paper awards.